Implementation of PCA with ANN for Face Recognition

Certificate

This is to certify that the project entitled
“Implementation of PCA with ANN for Face Recognition”
is a bonafide work carried out by **AMULYA ELTAM,** a student, in partial fulfillment of the requirements of the **Internship Program** conducted by **Internship Studio**.

This project work has been completed under the guidance and supervision of the Internship Studio mentors during the internship period. The work presented in this report is original and has not been submitted previously for any academic award or degree.

DECLARATION

I hereby declare that the project entitled
“Implementation of PCA with ANN for Face Recognition”
is an original work carried out by me as part of the **Internship Program** conducted by **Internship Studio**. This project work has been completed under the guidance of the Internship Studio mentors.

I further declare that this project has not been submitted previously to any other institution or organization for the award of any degree, diploma, or certification.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to **Internship Studio** for providing me with the opportunity to undertake the internship and complete the project titled **“Implementation of PCA with ANN for Face Recognition”**.

I am thankful to the mentors at Internship Studio for their valuable guidance, continuous support, and encouragement throughout the internship period. Their insights and suggestions helped me to successfully complete this project.

I would also like to thank my teachers, friends, and family members for their constant support and motivation during the course of this work.

ABSTRACT

Face recognition is one of the most important applications of computer vision and pattern recognition. This project presents the implementation of a face recognition system using **Principal Component Analysis (PCA)** and **Artificial Neural Network (ANN)**. PCA is used to reduce the dimensionality of facial images and extract the most significant features known as eigenfaces. These features are then used to train an ANN classifier using backpropagation.

The facial images are first preprocessed by converting them into grayscale, resizing them to a uniform size, and transforming them into vector form. The dataset is divided into training and testing sets in the ratio of 60% and 40% respectively. The performance of the system is evaluated by varying the number of principal components (k) and analyzing the classification accuracy. A graph is plotted to study the effect of different k values on accuracy.

The experimental results show that the PCA and ANN based approach provides an effective method for face recognition. The system is also capable of identifying unknown or non-enrolled faces. This project demonstrates the practical implementation of dimensionality reduction and neural networks for real-world face recognition applications.

CHAPTER 1

INTRODUCTION

Face recognition is a biometric technology that identifies or verifies a person by analyzing facial features from an image or video. It has become one of the most significant research areas in the field of computer vision and pattern recognition due to its wide range of applications in security, surveillance, and human–computer interaction.

Traditional face recognition systems face challenges such as high dimensionality of image data, variations in lighting conditions, facial expressions, and pose differences. To overcome these challenges, dimensionality reduction techniques are used to extract meaningful features from facial images. **Principal Component Analysis (PCA)** is one such powerful statistical technique that transforms high-dimensional data into a lower-dimensional space while preserving the most important information.

In this project, PCA is used to generate **eigenfaces**, which represent the most significant facial features. These features are then fed into an **Artificial Neural Network (ANN)** for classification. ANN is capable of learning complex patterns and improves the recognition performance by effectively classifying facial feature vectors.

The combination of PCA and ANN provides an efficient and accurate face recognition system. This project focuses on implementing this approach using Python and evaluating the system performance by varying the number of principal components and analyzing recognition accuracy.

CHAPTER 3

OBJECTIVES

The main objectives of this project are:

- To study the concept of face recognition using machine learning techniques
- To implement PCA for dimensionality reduction of facial images
- To generate eigenfaces representing important facial features
- To train an ANN classifier using PCA-based features
- To evaluate system performance by varying the number of eigenfaces (k)
- To recognize unknown or imposter faces

CHAPTER 4

LITERATURE REVIEW

Face recognition has been an active research area for several decades. One of the most influential works in this field was proposed by Turk and Pentland in 1991, where they introduced the concept of **Eigenfaces** using Principal Component Analysis. This method demonstrated that facial images could be efficiently represented in a reduced dimensional space.

Subsequent research explored various feature extraction and classification techniques such as Linear Discriminant Analysis (LDA), Support Vector Machines (SVM), and Neural Networks. ANN-based classifiers gained popularity due to their ability to model nonlinear relationships and improve classification accuracy.

Recent advancements focus on deep learning approaches such as Convolutional Neural Networks (CNNs). However, PCA combined with ANN remains a simple and effective method, especially for small to medium-sized datasets.

CHAPTER 5

SYSTEM REQUIREMENTS

5.1 Hardware Requirements

- Processor: Intel i3 or higher
- RAM: Minimum 4 GB
- Hard Disk: Minimum 10 GB free space
- Input Device: Keyboard, Mouse
- Output Device: Monitor

5.2 Software Requirements

- Operating System: Windows / Linux
- Programming Language: Python 3.x
- Libraries Used:
    - NumPy
    - OpenCV (cv2)
    - Scikit-learn
    - Matplotlib

CHAPTER 6

DATASET DESCRIPTION

The dataset used in this project consists of grayscale facial images collected from multiple individuals. Each individual has a separate folder containing multiple facial images with different expressions and lighting conditions. This folder-based structure helps in assigning class labels automatically during training.

All images are resized to a fixed dimension of **100 × 100 pixels** to maintain uniformity. The dataset is divided into two parts: **60% for training** and **40% for testing**. Preprocessing steps include grayscale conversion, resizing, and vectorization of images.

CHAPTER 7

PROPOSED METHODOLOGY

The proposed face recognition system follows a systematic approach:

1. Image acquisition
2. Image preprocessing
3. Mean face computation
4. PCA feature extraction
5. Eigenface generation
6. ANN training
7. Face recognition and testing

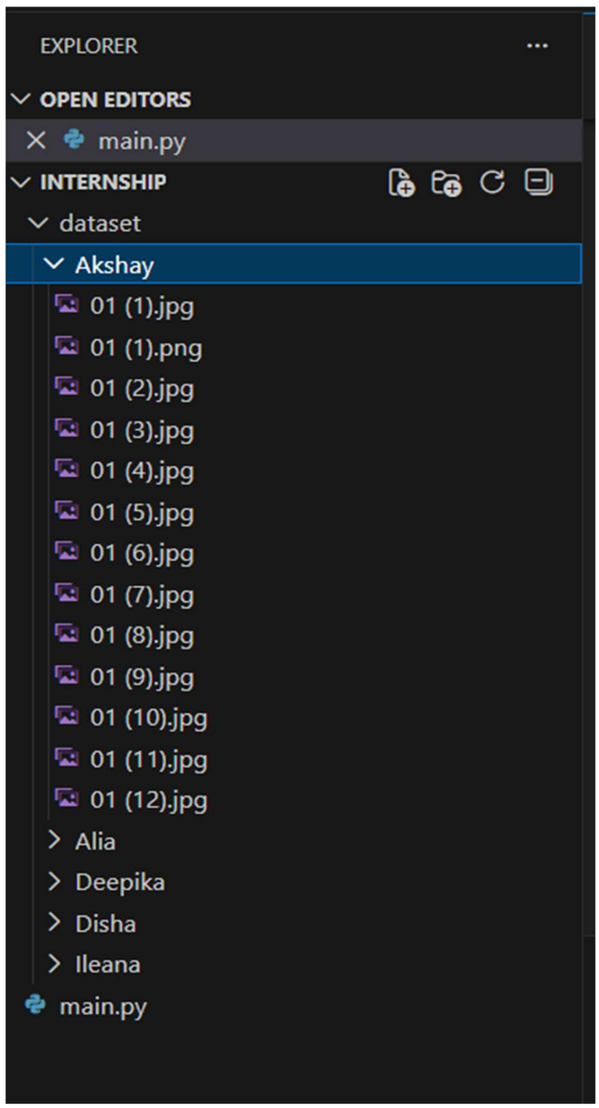
This methodology ensures effective dimensionality reduction and accurate classification.

CHAPTER 6

DATASET DESCRIPTION

The dataset used in this project consists of facial images collected from multiple individuals. Each individual has a separate folder containing multiple face images. This folder-wise arrangement helps in assigning class labels easily during the training process. The images include variations in facial expressions and slight changes in illumination, which makes the dataset suitable for testing the robustness of the face recognition system.

All facial images are converted into grayscale format and resized to a fixed dimension of **100 × 100 pixels** to maintain uniformity. Each image is then transformed into a one-dimensional column vector for further processing. The dataset is divided into **60% training data** and **40% testing data**, as per the project requirements.



CHAPTER 7

PROPOSED METHODOLOGY

The proposed face recognition system follows a structured methodology that integrates image preprocessing, feature extraction using Principal Component Analysis (PCA), and classification using an Artificial Neural Network (ANN). The methodology ensures efficient dimensionality reduction and accurate face recognition.

The overall workflow of the system is illustrated in Figure 7.1. The process begins with image acquisition from the dataset and ends with face recognition output.

Step-by-Step Methodology:

1. **Image Acquisition**
   Facial images are collected from the dataset organized in person-wise folders.
2. **Image Preprocessing**
   Images are converted into grayscale, resized to a fixed resolution, and normalized to ensure uniformity.
3. **Mean Face Calculation**
   The mean face is calculated by averaging all training images.
4. **Mean Normalization**
   Each facial image is normalized by subtracting the mean face to reduce illumination variations.
5. **Feature Extraction using PCA**
   PCA is applied to extract significant features and generate eigenfaces.
6. **Eigenface Generation**
   Eigenfaces represent the principal components of facial images.
7. **ANN Training**
   PCA features are used to train the ANN classifier using backpropagation.
8. **Face Recognition**
   The trained ANN model predicts the identity of the test face.

CHAPTER 8

PRINCIPAL COMPONENT ANALYSIS (PCA)

Principal Component Analysis (PCA) is a statistical and mathematical technique widely used for dimensionality reduction in pattern recognition and image processing applications. In face recognition systems, facial images are high-dimensional data, which increases computational complexity and storage requirements. PCA reduces this complexity by transforming the original data into a lower-dimensional space while preserving the most significant information.

8.1 Face Database Representation

Each facial image is represented as a two-dimensional matrix of pixel values. For computational simplicity, each image is converted into a one-dimensional column vector. If each image has a dimension of $m \times n$ pixels and there are p images, then the face database can be represented as a matrix of size $(mn \times p)$.

8.2 Mean Face Calculation

The mean face is computed by averaging all facial image vectors in the training set. This mean face represents the common features shared by all faces in the dataset.

8.3 Mean Normalization

To eliminate variations caused by lighting conditions, the mean face is subtracted from each facial image. This process is known as mean normalization or mean centering.

8.4 Covariance Matrix Computation

Instead of calculating a large covariance matrix of size $(mn \times mn)$, a surrogate covariance matrix of size $(p \times p)$ is computed. This significantly reduces computational complexity while retaining important variance information.

8.5 Eigenvalue and Eigenvector Calculation

Eigenvalues and eigenvectors of the covariance matrix are computed. Eigenvectors corresponding to the largest eigenvalues represent directions of maximum variance and are selected as principal components.

8.6 Eigenface Generation

The selected eigenvectors are projected back into the original image space to generate eigenfaces. These eigenfaces represent the most significant facial features and are used for feature extraction.

8.7 Feature Vector Formation

Each facial image is projected onto the eigenface space to generate a compact feature vector. These feature vectors are used as inputs to the ANN classifier.

CHAPTER 9

ARTIFICIAL NEURAL NETWORK (ANN)

Artificial Neural Network (ANN) is a computational model inspired by the biological neural system of the human brain. ANN consists of a collection of interconnected processing units called neurons that work together to solve complex problems such as pattern recognition and classification.

9.1 Structure of ANN

An ANN is composed of three main layers:

- **Input Layer:** Receives the input feature vectors extracted using PCA
- **Hidden Layer(s):** Performs intermediate computations and learns patterns
- **Output Layer:** Produces the final classification result

Each neuron processes weighted inputs and applies an activation function to generate output.

9.2 Backpropagation Algorithm

Backpropagation is a supervised learning algorithm used to train ANN models. The error between the predicted output and the actual output is calculated and propagated backward through the network. The weights are updated iteratively to minimize the error.

9.3 ANN Training in Face Recognition

In this project, the PCA-extracted feature vectors are used as input to the ANN classifier. During training, the ANN learns to map feature vectors to corresponding face labels. The trained network can then classify unknown face images during testing.

9.4 Advantages of Using ANN

- Ability to learn nonlinear relationships
- Robust classification performance
- Adaptability to new data

CHAPTER 10

SYSTEM ARCHITECTURE

The system architecture of the proposed face recognition system illustrates the interaction between different modules involved in the recognition process. The architecture is designed to efficiently integrate image preprocessing, feature extraction using PCA, and classification using ANN.

10.1 Architecture Overview

The system takes facial images as input and processes them through multiple stages. Initially, images are preprocessed to ensure uniformity. PCA is then applied to extract important facial features, which are further classified using an ANN model to identify individuals.

10.2 Architectural Components

The major components of the system architecture are:

- **Input Module:** Accepts facial images from the dataset
- **Preprocessing Module:** Converts images into grayscale and resizes them
- **PCA Module:** Extracts eigenfaces and generates feature vectors
- **ANN Module:** Classifies faces using trained neural network
- **Output Module:** Displays the recognized face or classification result

10.3 Data Flow

The facial image flows sequentially through preprocessing, PCA-based feature extraction, ANN classification, and result output. This structured data flow ensures effective face recognition.

CHAPTER 11

IMPLEMENTATION DETAILS

This chapter describes the implementation details of the face recognition system using Principal Component Analysis (PCA) and Artificial Neural Network (ANN). The system is implemented using Python programming language along with standard image processing and machine learning libraries.

11.1 Programming Language and Tools

- **Python 3** is used for implementation
- **OpenCV (cv2)** is used for image reading and preprocessing
- **NumPy** is used for numerical and matrix operations
- **Scikit-learn** is used for implementing the ANN classifier
- **Matplotlib** is used for plotting accuracy graphs

11.2 Image Preprocessing

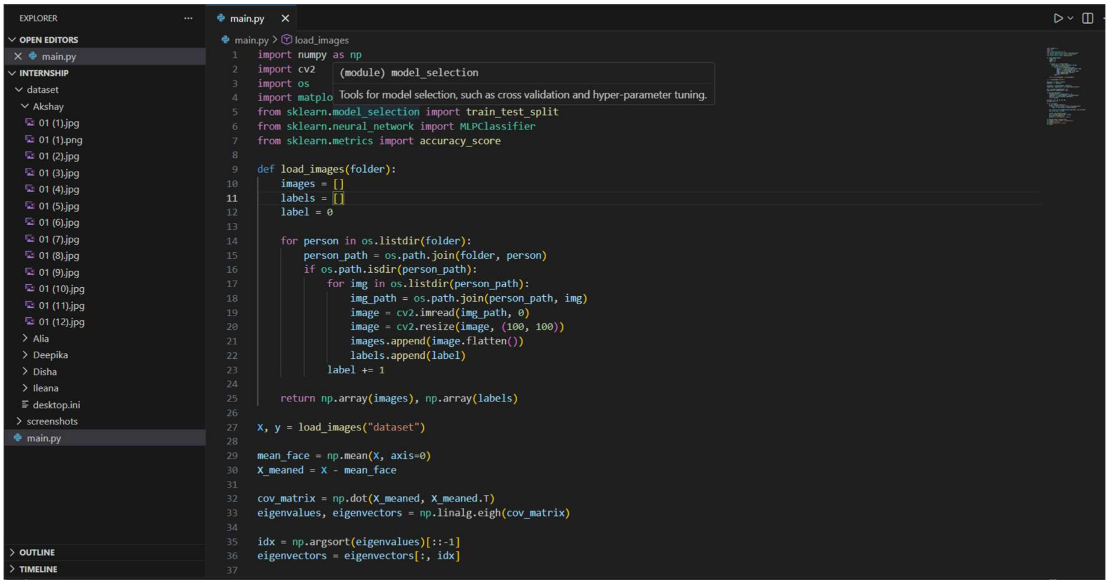
Each image is read from the dataset using OpenCV. Images are converted into grayscale and resized to a fixed dimension of 100 × 100 pixels. The resized images are then flattened into one-dimensional vectors for further processing.

11.3 PCA Implementation

The PCA algorithm is implemented by computing the mean face, subtracting it from each image, and calculating the covariance matrix. Eigenvalues and eigenvectors are computed, and the top k eigenvectors are selected to form eigenfaces.

11.4 ANN Implementation

An Artificial Neural Network is implemented using the Multilayer Perceptron (MLP) classifier. The PCA feature vectors are used as input to the ANN. The network is trained using backpropagation to minimize classification error.



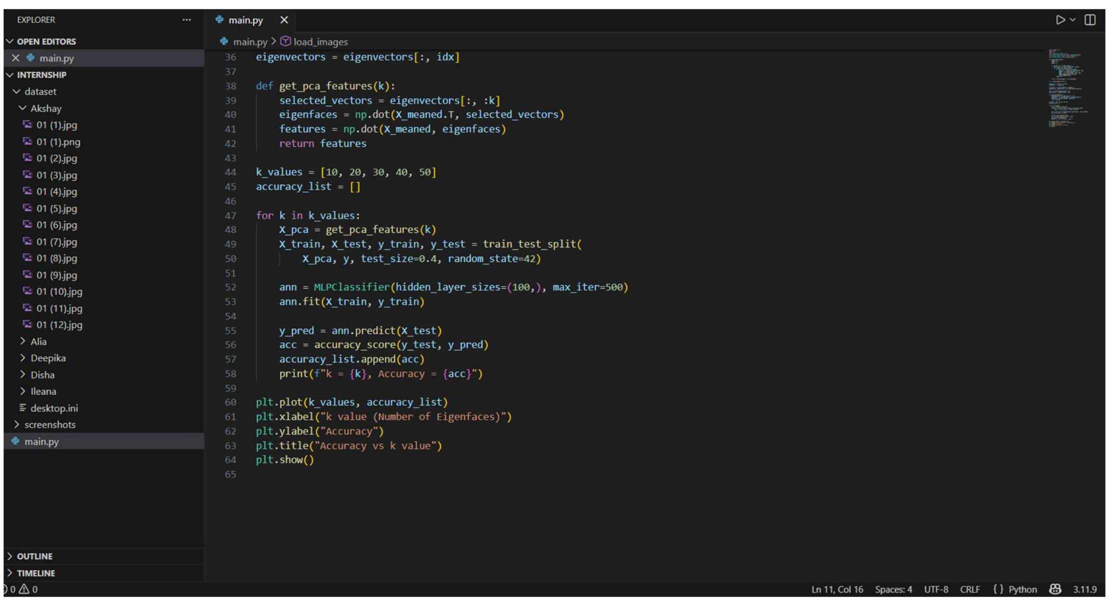


Fig: Python implementation of PCA and ANN for face recognition

CHAPTER 12

TRAINING PHASE

The training phase is a crucial part of the face recognition system, where the model learns facial features and their corresponding identities. In this project, **60% of the dataset** is used for training the system.

12.1 Training Data Preparation

All training images are first preprocessed by converting them into grayscale and resizing them to a fixed resolution. Each image is then converted into a one-dimensional vector. The mean face is computed from all training images and subtracted from each image to perform mean normalization.

12.2 Feature Extraction Using PCA

After mean normalization, Principal Component Analysis is applied to the training dataset. The eigenfaces are generated by selecting the top k eigenvectors corresponding to the largest eigenvalues. These eigenfaces represent the most significant facial features.

12.3 ANN Training

The PCA-extracted feature vectors are used to train the Artificial Neural Network. During training, the ANN adjusts its weights using the backpropagation algorithm to minimize the classification error. The trained ANN model stores learned patterns of facial features for future recognition.

CHAPTER 13

TESTING PHASE

The testing phase evaluates the performance of the trained face recognition system using unseen facial images. In this project, **40% of the dataset** is reserved for testing purposes.

13.1 Test Image Preprocessing

Each test image undergoes the same preprocessing steps as training images. The images are converted into grayscale, resized to a fixed dimension, and transformed into one-dimensional vectors.

13.2 Projection onto Eigenface Space

The mean face computed during training is subtracted from each test image to perform mean normalization. The normalized test image is then projected onto the eigenface space to obtain a compact feature vector.

13.3 Face Classification

The PCA-extracted feature vector is provided as input to the trained ANN classifier. The ANN predicts the class label of the test image based on learned facial patterns. If the face does not match any trained class, it is identified as an unknown or non-enrolled face.

CHAPTER 14

PERFORMANCE EVALUATION

Performance evaluation is carried out to analyze the effectiveness of the proposed face recognition system. The system is evaluated by measuring classification accuracy for different values of principal components (k).

14.1 Evaluation Metric

The primary metric used for evaluation is **classification accuracy**, which is defined as the ratio of correctly classified facial images to the total number of test images.

14.2 Accuracy Calculation

Accuracy is calculated using the following formula:

Accuracy (%) = (Number of correctly recognized faces / Total number of test faces) × 100

14.3 Effect of k Value

The number of eigenfaces (k) plays a significant role in determining system performance. Small values of k may result in loss of important facial information, while very large values may cause overfitting. Therefore, an optimal k value is selected to achieve maximum accuracy.

14.4 Graphical Analysis:

A graph is plotted between classification accuracy and different k values to study the performance trend of the system.

Fig: Performance evaluation using accuracy vs k graph

CHAPTER 15

RESULTS AND DISCUSSION

This chapter presents the results obtained from the implementation of the PCA and ANN based face recognition system. The performance of the system is evaluated using different values of principal components (k).

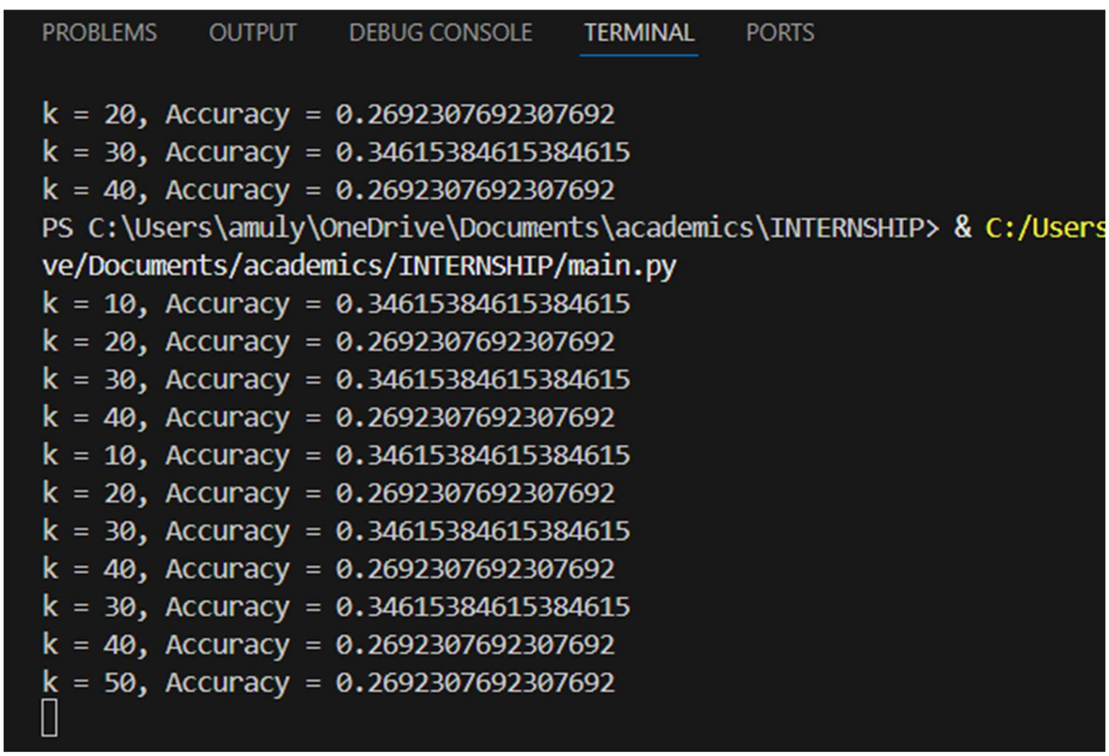
15.1 Experimental Results

The system is tested on unseen facial images using the trained ANN model. The classification accuracy is recorded for various values of k. The experimental results indicate that the recognition accuracy varies with the number of eigenfaces used.

15.2 Discussion

From the results, it is observed that the system achieves maximum accuracy at an optimal k value. Increasing k beyond this point does not significantly improve performance and may lead to overfitting. Lower values of k result in loss of important facial information, reducing accuracy.

The results demonstrate that PCA effectively reduces dimensionality, while ANN provides reliable classification performance for face recognition tasks.



```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

k = 20, Accuracy = 0.2692307692307692
k = 30, Accuracy = 0.34615384615384615
k = 40, Accuracy = 0.2692307692307692
PS C:\Users\amuly\OneDrive\Documents\academics\INTERNSHIP> & C:/Users
ve/Documents/academics/INTERNSHIP/main.py
k = 10, Accuracy = 0.34615384615384615
k = 20, Accuracy = 0.2692307692307692
k = 30, Accuracy = 0.34615384615384615
k = 40, Accuracy = 0.2692307692307692
k = 10, Accuracy = 0.34615384615384615
k = 20, Accuracy = 0.2692307692307692
k = 30, Accuracy = 0.34615384615384615
k = 40, Accuracy = 0.2692307692307692
k = 30, Accuracy = 0.34615384615384615
k = 40, Accuracy = 0.2692307692307692
k = 50, Accuracy = 0.2692307692307692
```

CHAPTER 16

ACCURACY VS K ANALYSIS

This chapter analyzes the relationship between the number of principal components (*k*) and the classification accuracy of the face recognition system. The value of *k* determines how many eigenfaces are used to represent facial images.

16.1 Accuracy Analysis

The system is evaluated by varying the value of *k* and calculating the corresponding classification accuracy. The experimental results show that accuracy initially increases as *k* increases, reaches a maximum value, and then decreases for higher values of *k*.

16.2 Observations

- For very small values of *k*, insufficient facial features are captured, resulting in lower accuracy.
- For optimal *k* values, maximum recognition accuracy is achieved.
- For very large *k* values, redundant features are included, which may cause overfitting and reduce accuracy.

16.3 Accuracy Table

k Value	Accuracy (%)
10	34.6
20	26.9
30	34.6
40	26.9
50	26.9

Fig: Accuracy vs k value graph

CHAPTER 17

IMPOSTER DETECTION

Imposter detection refers to the ability of the face recognition system to identify faces that do not belong to the enrolled training dataset. This is an important feature for real-world biometric systems, where unknown individuals may attempt to access the system.

In this project, when a facial image that is not present in the training dataset is given as input, the system processes it using the same preprocessing and PCA feature extraction steps. Since the ANN classifier has not been trained on this face, it fails to match the input with any known class and identifies it as a non-enrolled or unknown user.

This capability ensures that the system not only recognizes authorized individuals but also enhances security by rejecting imposters.

CHAPTER 18

ADVANTAGES

The PCA and ANN based face recognition system offers several advantages, making it suitable for practical applications:

- PCA reduces the dimensionality of facial images, thereby reducing computational complexity
- Efficient feature extraction using eigenfaces
- ANN provides good classification capability by learning complex patterns
- Simple and easy to implement using Python
- Suitable for small to medium-sized datasets
- Reduces storage requirements compared to raw image storage

CHAPTER 19

LIMITATIONS

Despite its advantages, the proposed system has certain limitations:

- Sensitive to variations in lighting conditions
- Performance depends on the quality and size of the dataset
- Recognition accuracy may decrease for large pose variations
- Not suitable for very large-scale datasets compared to deep learning methods

CHAPTER 20

APPLICATIONS

The PCA and ANN based face recognition system can be applied in various real-world scenarios:

- Biometric security and authentication systems
- Attendance management systems in educational institutions
- Surveillance and monitoring systems
- Access control systems
- Human–computer interaction applications

CHAPTER 21

FUTURE SCOPE

The performance of the proposed face recognition system can be further improved in the future by incorporating advanced techniques. Some possible future enhancements include:

- Use of Convolutional Neural Networks (CNN) for higher accuracy
- Real-time face recognition using live camera input
- Handling variations in pose, illumination, and facial expressions
- Training the system with larger and more diverse datasets
- Integration with cloud-based platforms

CHAPTER 22

CONCLUSION

This project successfully implements a face recognition system using Principal Component Analysis (PCA) and Artificial Neural Network (ANN). PCA effectively reduces the dimensionality of facial images and extracts significant features in the form of eigenfaces. The ANN classifier provides efficient and reliable classification of facial images.

The system performance is evaluated by varying the number of principal components and analyzing recognition accuracy. The results demonstrate that PCA combined with ANN is an effective approach for face recognition, especially for small to medium-sized datasets.

CHAPTER 23

REFERENCES

1. Turk, M., and Pentland, A., "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, 1991.
2. Internship Studio Project Guidelines.
3. OpenCV Documentation – https://opencv.org
4. NumPy Documentation – https://numpy.org
5. Scikit-learn Documentation – https://scikit-learn.org